REMARKS/ARGUMENTS

Applicant has canceled all pending claims and added new claims 40-88.

The Examiner rejects claims 2 and 27 provisionally under 35 U.S.C.§101 as claiming the same invention as that of claims 2 and 26 of copending Application No. 09/921,832.

This rejection is moot in light of the cancellation of claims 2 and 26.

The Examiner provisionally rejects claims 9 and 16 under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 8 and 17 of copending Application 09/921,832 in view of Jordan (U.S. Published Application 2002/0026560).

This rejection is moot in light of the cancellation of claims 9 and 16.

The Examiner rejects claims 32-35 under 35 U.S.C.§102(e) as being anticipated by Jordan et al.; claims 36-37 under 35 U.S.C.§102(e) as being anticipated by Apostolopoulos et al.; Claims 1-3, 5, 7-9, 14, 16-18, 23, 25-28 and 30 under 35 U.S.C.§103(a) as being unpatentable over Hankinson et al. (U.S.6,799,202) in view of Schmiedler et al. (U.S. 6,763,370); claims 4 and 29 under 35 U.S.C.§103(a) as being unpatentable over Hankinson and Schmiedler and further in view of Masters (U.S. Published Application 2002/0040400); claims 6 and 31 under 35 U.S.C.§103(a) as being unpatentable over Hankinson and Schmiedler as applied to claims 1 and 26 above and further in view of Logue et al. (U.S. 6,330,606); claims 10-12, 15, 19-21, and 24 under 35 U.S.C.§103(a) as being unpatentable over Hankinson and Schmiedler as applied to claim 7 and further in view of Jordan; claims 13 and 22 under 35 U.S.C.§103(a) as being unpatentable over Hankinson and Schmiedler as applied to claims 7 and 16 and further in view of Masters (U.S. Published Application 2002/0040400)

and Jordan; and claims 34 under 35 U.S.C.§103(a) as being unpatentable over Jordan in view of Ross, K. "Hash Routing for Collections of Shared Web Catches".

Applicants respectfully traverse the Examiner's rejections. The cited references fail to teach or suggest at least the following italicized features in the independent claims:

> 40. An arrangement for serving information requests, comprising:
>
> a plurality of information servers connected to a communications network, all of the information servers having a common address on the communications network and serving a set of information to clients, each of the information servers being configured to receive a transaction request associated with an individual transaction and to provide a response to each transaction request; and
>
> a content director connecting the information servers to the communications network and distributing transaction requests among the information servers comprising:
>
> a flow switch that selects an appropriate information server to service each transaction request and thereafter forwards at least portions of the transaction request to a selected one of the information servers;
>
> a cache that stores, *in a hot invariant table*, a plurality of objects corresponding to at least some of the transaction requests forwarded to one or more of the information servers, *the hot invariant table identifying information frequently requested from the information servers and including, for each invariant identifying corresponding information, a hit counter indicating a number of transaction requests, received over a determined time interval, requesting the corresponding information;*
>
> *a cache processor that accesses the plurality of objects in response to communications received from the flow switch;*
>
> *a digest generator that generates, when the hit counter for an invariant indicates at least a threshold transaction request receipt frequency, a digest value pointing to the location in the hot invariant table where the corresponding entry is stored*; and
>
> *a digest store that stores the digests corresponding to frequently requested content.*
>
> 54. In an arrangement comprising a plurality of information servers connected to a communications network, each of the information servers being configured to receive a transaction request associated with an individual transaction

and to provide a response to each transaction request, a method for serving transaction requests from clients comprising:

maintaining a hot invariant table identifying information frequently requested from the information servers, the hot invariant table including, for each invariant identifying corresponding information, a hit counter indicating a number of transaction requests, received over a determined time interval, requesting the corresponding information;

generating, when the hit counter for a selected invariant indicates at least a threshold transaction request receipt frequency, a digest value pointing to the location in the hot invariant table where the entry corresponding to the selected invariant is stored; and

accessing a digest store comprising the digest values to select an information server to service a transaction request for frequently requested information.

69. An arrangement for serving information requests, comprising:

a plurality of information servers connected to a communications network, all of the information servers having a common address on the communications network and serving a set of information to clients, each of the information servers being configured to receive a transaction request associated with an individual transaction and to provide a response to each transaction request; and

a content director connecting the information servers to the communications network and distributing transaction requests among the information servers comprising:

first flow switching means for parsing plain text transaction requests to locate selected fields, selecting an appropriate information server to service each transaction request, and thereafter forwarding at least portions of the parsed transaction requests to a selected one of the information servers;

cache means for storing, *in a hot invariant table*, a plurality of objects corresponding to transaction requests forwarded to one or more of the information servers, *the hot invariant table identifying information frequently requested from the information servers and including, for each invariant identifying corresponding information, a hit counter indicating a number of transaction requests, received over a determined time interval, requesting the corresponding information;*

cache processing means for accessing the plurality of objects in response to communications received from the first flow switching means;

digest generator means for generating, when the hit counter for an invariant indicates at least a threshold transaction request receipt frequency, a digest value pointing to the location in the hot invariant table where the corresponding entry is stored; and

digest store means for storing the digests corresponding to frequently requested content.

75. In an arrangement comprising a plurality of information servers connected to a communications network, each of the information servers being configured to receive a transaction request associated with an individual transaction and to provide a response to each transaction request, a method for serving transaction requests from clients comprising:

maintaining a hot invariant table identifying information frequently requested from the information servers, the hot invariant table including, for each invariant identifying corresponding information, a hit counter indicating a number of transaction requests, received over a determined time interval, requesting the corresponding information;

when the hit counter for an invariant indicates at least a threshold transaction request receipt frequency, locating the information associated with the invariant at a cache information server and thereafter directing a transaction request for information associated with the invariant to a cache information server; and

when the hit counter for an invariant falls below a threshold transaction request receipt frequency, directing a transaction request for information associated with the invariant to an origin information server.

In one embodiment, the present invention is directed to a number of features of an OSI Layer 2, 3, or 4 Web switch (or intelligent flow switch) in a server farm.

In one of the features, the switch maintains a hot invariant table, a current connection table, and a digest store. The hot invariant table identifies "hot" or frequently requested content and includes source invariants (e.g., URLs), destination invariants (e.g., URLs), cookie name and value, a timestamp indicating when an entry was last updated, a hit counter, and/or a tag (generated by the switch when in the tagging mode). The current connection table maintains a record of all current virtual circuits for wire speed packet switching. The digest store includes, for each "hot" destination invariant (e.g., URL and/or and/or cookie value), a digest value. The digest value is computed by

a hashing function (Specification at p. 15) and acts as a pointer to a memory location of a corresponding object (e.g., URL) in the hot invariant table. The digests of other switches in the server farm are shared among switches to facilitate identification of the appropriate switch serving desired content. When a "hit counter" associated with specific content (e.g., URL) exceeds a selected threshold, the content is considered to be "hot" and the content is thereafter served from a cache server rather than an origin server and a digest for the "hot content" is stored in the digest store. When content is no longer "hot", it is no longer served from a cache server but from the origin server and its digest is removed from the digest store.

In another feature, a transaction request or response is tagged with both a cookie generated by an assigned information server (which is known) and a tag associated with the information server (which is inventive). The tag identifies a cache or origin server currently serving content. The bit size of the tag is generally much smaller than that of the cookie.

In another feature, the network switch operates in at least two modes of operation, namely tagging and digesting modes. In the tagging mode, tags identifying an assigned information server are generated and appended to transaction requests and/or responses. In the digesting mode, digests are generated, content hotness is monitored, and transaction requests are routed to servers based on hotness and/or cookie. Alternatively, the two modes can operate concurrently rather than discretely.

In yet another feature, encrypted packets are decrypted before the packet is processed by the intelligent flow switch.

Jordan et al.

Jordan is directed to a system including a collection of cooperating cache servers, such as proxy cache servers. A request is forwarded to a cooperating cache server if the requested objects cannot be found locally. An overload condition is detected some objects are in high demand by all the clients, and cache servers that contain those hot objects become overloaded due to forwarded requests. An overload condition is detected by monitoring the weighted sum of a count of the forwarded requests and a count of direct requests to the cooperating cache server. The load monitor can include a logical directory server for maintaining a load table for monitoring the load on the cache servers and a caching table (or hash function) for monitoring the forwarding frequency and locating objects. In response, the load is balanced by shifting some or all of the forwarded requests from an overloaded cache server to a less loaded one. Jordan teaches locating an object in a cooperating cache server using a hash function applied to a URL or destination IP address. This reference fails to teach, *inter alia,* the use of a hot table to effect content relocation in memory rather than load balancing; the use of a digest to point to a corresponding location of a URL in the hot table; and directing content requests having no corresponding digest value to an origin server and content requests having a corresponding digest value in the digest store to a cache server. In Jordan, the hashing function is used to effect load balancing, *i.e.,* identify a server which should service a request and redirect the request to the server.

Apostolopoulos Article

The Apostolopoulos article discloses hashing on the URL to map URLs to a cache/server. However, it discourages this approach stating that "[a] drawback of this approach is that the hash function assigns any given URL to a single Web cache. If there are some 'hot' pages that are accessed very frequently then this scheme can result in a rather poor load distribution as most of the requests will be routed to a single cache in the cluster." (page 1123, Chapter B.1) The reference fails to disclose the maintenance of a hot table of URLs, with the location of the URLs in the table being related to the magnitudes of the digest values. Moreover, the reference fails to teach tagging by the network or Web switch in addition to tagging by the various information servers.

Hankinson

Hankinson et al. is directed to a high speed server farm in which different functions of the server's state machine are distributed across a plurality of processors running a plurality of operating systems. The web server has a number of members categorized into member classes. Each member class has a distinct specialized operating system that is optimized for its function. Together, the operating systems of the members make up an operating system referred to as the Federated Operating System. Receiver members receive requests from clients over the external network and pass off data from requests to dispatcher members over the internal network. A dispatcher member uses the internal network to send information to a responder member, instructing the responder member to send data requested by the client to the client over the external network. Load balancing

(such as is performed by a traffic manager prior to routing by the flow switch) is performed without regard to the message contents prior to transmission of a message to a dispatcher 720 (col. 17, lines 41-50).

With reference to Fig. 7 and col. 21, line 64-col. 22, line 8, an encrypted message is transferred from a receiver 745 (or input) to a dispatcher (or switch) 720. Dispatcher 720 then sends the message to another dispatcher 725 over a private connection 730. Dispatcher 725 then sends the message to a decoder 735, which decodes the message and returns the decoded message to dispatcher 725. Dispatcher 725 then sends a message identifying the location of the requested data to one of responders 740, and the responder 740 retrieves and sends the information to a decoder 735 for encryption and subsequently forwards an encrypted response, containing the encrypted information, to the client.

As conceded by the Examiner, Hankinson does not teach a digest generator that generates a digest based on packet header information, with the digest corresponding to a location in the cache where an object corresponding to a transaction is stored. (Office Action at page 7.)

Schmiedler

Schmeidler et al. is directed to a system for secure delivery of on-demand content over broadband access networks that uses servers and security mechanisms to prevent client processes from accessing and executing content without authorization. A plurality of encrypted titles are stored on a content server coupled to the network. An access server also coupled to the network contains

the network addresses of the titles and various keying and authorization data necessary to decrypt and execute a title. A client application executing on a user's local computer system is required to retrieve the address, keying, and authorization data from the access server before retrieving a title from the content server and enabling execution of the title on a user's local computer system. To effect retrieval, a briq is mapped into a directory and file where it is stored in memory. In this manner, file system 1008 functions as an interface between the network request from the SCDP system and the memory 1050.

Fig. 12 diagrams a briq. A briq 1200 includes a briq header 1202, a cryptoblock 1204, a superblock 1206, and one or more titles 1208A-N. A URN is a unique identifier of a title within a briq. The URN can correspond exactly to the current location of the title in the vendor's storage server. A URL identifies the current location of the briq in a RAFT storage server.

While Schmeidler et al. does disclose the use of a hashing function at col. 18, lines 44-51, the hash code and an encryption key are used to digitally sign a launch string and not as a content locator. The hash code is not related to a stored location of an object.

Masters

Masters is directed to a method and system for testing and examining cookies in the data streams of HTTP connections to persistently direct HTTP connections to the same destination. The system enables a network device to direct subsequent HTTP connections from the same client to the same server (destination) for accessing the requested resources. There are four modes for employing

the cookie to persistently direct HTTP connections. The associative mode inserts a cookie that uniquely identifies the client into an HTTP response. The passive mode inserts cookie information that uniquely identifies a previously selected destination into an HTTP response. In the rewrite mode, a network device manages the destination information that is rewritten over blank cookie information generated by the destination producing the HTTP response. The insert mode inserts and removes cookie information in the data packets for HTTP requests and responses prior to processing by the destination.

Logue

Logue et al. is directed to a method and apparatus for dispatching document requests in a proxy to more efficiently allocate the document cache space within a proxy. A proxy includes a document cache storing recently requested documents. The proxy is coupled to a client and a remote server. A URL is included in the document request. The proxy forwards the request to one of a plurality of proxy servers based on the URL. The proxy may perform a hash function on the URL that maps the URL to exactly one of the plurality of proxy servers. Although Logue et al. teaches tracking content hits, the hit counters are used not for hot content identification and content relocation but for administrative purposes.

The pending independent claims are therefore allowable over the cited references.

The dependent claims provide further reasons for allowability over the cited references.

By way of example, dependent claims 41, 55, 70, and 76 are directed to a cryptographic module that decrypts, prior to parsing and information server selection by the flow switch, cipher text

transaction requests and provides plain text transaction requests to the flow switch, wherein, prior to decryption, the cipher text transaction requests have not been routed by another flow switch. Dependent claims 42, 56, 71, and 77 are directed to the simultaneous or near simultaneous receipt of encrypted transaction requests from different clients having a common electronic address on the network. Hankinson et al. teaches the routing of the encrypted message first from a receiver 745 (which performs an initial analysis of the message col. 29, lines 33-37) to a first dispatcher 720 and second from a first dispatcher 720 to a second dispatcher 725 *before the message is decrypted.* In contrast, the claimed invention decrypts the message before it is initially routed by a switch, such as the receiver. Accordingly, Hankinson et al. fails to address the problem noted above, namely how to distinguish transaction requests from different clients having a common address on the communications network.

Dependent claims 47, 61, 73, and 81 are directed to the specific hashing algorithm employed by an embodiment of the claimed invention.

Dependent claims 43, 50-52, 57, 64-66, 74, 80, and 84-86 are directed to the use of tags (other than server-generated cookies) to direct transaction requests from clients. Masters does not teach tagging a transaction request or response with *both* a cookie generated by an assigned information server *and a tag* associated with the information server.

Dependent claims 53, 67, and 87 are directed to the use of discrete operational modes by the flow switch. Specifically, during a first time interval, the flow switch is in a tagging mode in which the switch generates and appends tags to transaction responses and, during a second different time

interval, the switch is in a digesting mode in which digests are generated, invariant hotness is monitored, and transaction requests are routed to information servers based on requested invariant hotness and/or cookie.

Applicant wishes to clarify the intended meaning of certain claim language in light of the Federal Circuit decision "SuperGuide Corporation v. DirecTV Enterprises, Inc., et al., 358 F.3d 870 (Fed. Cir. 2004). In that decision, the Federal Circuit held, under the unique facts of that case, that the phrase "at least one of a desired program start time, a desired program end time, a desired program service, and a desired program type" means "at least one of a desired program start time, at least one of a desired program end time, at least one of a desired program service, and at least one of a desired program type".

Applicant has used the phrases "at least one of", "one or more", and "and/or" in a number of claims and wishes to clarify to the Examiner the proper construction of this phrase. Applicant intended the phrases "at least one", "one or more", and "and/or" as used in the claims to be an open-ended expression that is both conjunctive and disjunctive in operation. For example, the expressions "at least one of A, B and C", "one or more of A, B, and C", and "A, B, and/or C" mean A alone, B alone, C alone, A and B together, A and C together, B and C together, and A, B and C together. Applicant believes that this construction is consistent with the Examiner's construction of the claims in the Office Action. If the Examiner disagrees with this construction, Applicant respectfully requests that the Examiner notify Applicant accordingly so that Applicant can further amend the claims.

Based upon the foregoing, Applicants believe that all pending claims are in condition for allowance and such disposition is respectfully requested. In the event that a telephone conversation would further prosecution and/or expedite allowance, the Examiner is invited to contact the undersigned.

Respectfully submitted,

SHERIDAN ROSS P.C.

By: Douglas Swartz

Douglas W. Swartz
Registration No. 37,739
1560 Broadway, Suite 1200
Denver, Colorado 80202-5141
(303) 863-9700

Date: March 24, 2005